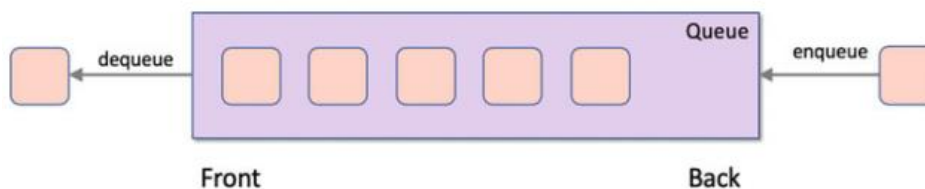


Лабораторийн ажил 1. Өмнөх мэдлэгээ бататгах даалгавар

Өгөгдлийн хийсвэл төрөл (ADT) нь нэг буюу хэд хэдэн тодорхой өгөгдлийн бүтцээр хэрэгжүүлж болох нийтлэг ойлголтыг тодорхойлоход хэрэглэгддэг. Эдгээр хэрэгжүүлэлтүүд нь зан төлвийг тодорхойлоход өгөгдлийн өөр дотоод дүрслэл эсвэл өөр алгоритмуудыг ашиглаж болно; семантик байдлаар ADT-ийн өгсөн тодорхойлолттой тохирч байдаг. Жишээлбэл, жагсаалт ADT нь өгөгдлийн бүтцийн нэгэн адил өгөх ёстой үйл ажиллагаа, зан төлвийг тодорхойлсон жагсаалтыг тодорхойлж болно. Хатуу хэрэгжилт нь элементүүдийн суурь массивыг ашиглан эсвэл элементүүдийг заагчтай холбож эсвэл хэш хүснэгтийн зарим хэлбэрийг ашиглан жагсаалтын семантиктай нийцэж болно (эдгээр нь жагсаалтыг хэрэгжүүлэхэд ашиглаж болох өөр дотоод дүрслэлүүд юм).

Жишээ дасгал. Дарааллыг классаар дүрслэх



```
class Queue:
def __init__(self):
self._list = [] # анхны утга

def enqueue(self, element):
self._list.append(element)

def dequeue(self):
return self._list.pop(0)

def __len__(self):
return len(self._list)

def is_empty(self):
return self.__len__() == 0

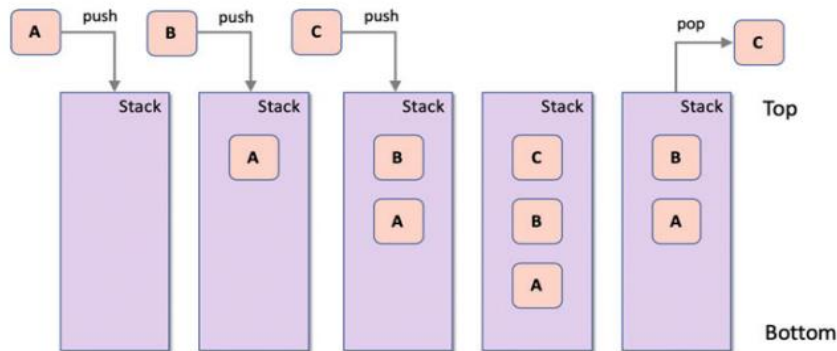
def peek(self):
return self._list[0]

def __str__(self):
return 'Queue: ' + str(self._list)
```

Жишээ турших

```
queue = Queue()
queue.enqueue('T1')
queue.enqueue('T2')
queue.enqueue('T3')
print('queue:', queue)
print('queue.is_empty():', queue.is_empty())
print('queue.len():', queue.len())
print('queue.peek():', queue.peek())
print('queue.dequeue():', queue.dequeue())
print('queue:', queue)
```

Дасгал 1. Стекийг жагсаалтаар дүрслэнэ үү.



Өөрийн Stack классыг хэрэгжүүл.

- Стек дээр элемент нэмэхэд ашигладаг push (element) арга.
- Stack-ийн дээд элементийг авах pop () арга (энэ арга тэр элементийг стекээс хасдаг).
- Стек дээрх дээд элементийг харах боломжийг олгодог top () арга (элементийг стекээс гаргахгүй байх).
- Стекийн хэмжээг буцаах __len__ () арга.
- Стек хоосон эсэхийг шалгадаг is_empty () арга.
- Стекийг мөр болгон хөрвүүлэхэд ашигладаг __str__ () арга.

Классыг бичсэний дараа дараах туршилтын программыг ажиллуулах боломжтой байх ёстой:

```
stack = Stack()
stack.push('T1')
stack.push('T2')
stack.push('T3')
print('stack:', stack)
print('stack.is_empty():', stack.is_empty())
print('stack.length():', stack.length())
print('stack.top():', stack.top())
```

```
print('stack.pop():', stack.pop())  
print('stack:', stack)
```

Дасгал 2. Дарааллыг collections сан ашиглан дүрслэх

```
from collections import deque
```

Өгөгдсөн утгуудыг зөв эрэмбэлсэн дараалал байхаар оруулах Python програм бич.